

Service-Centric Interoperability Model for Inter-Enterprise Applications

George Feuerlicht and Sooksathit Meesathit
Faculty of Information Technology,
University of Technology, Sydney
jiri@it.uts.edu.au

Abstract

Interoperability is a key prerequisite for the automation of electronic business activities across organizational boundaries. Existing approaches use document interchange as the interoperability mechanism for inter-enterprise applications and suffer from limited scalability and flexibility. In this paper we discuss the limitations of the document-centric interoperability model and then contrast it with the service-centric approach. We argue that the emergence of service oriented computing presents an opportunity to significantly improve the interoperability of electronic business applications by using well-designed standard service interfaces.

1. Introduction

Interoperability is a key prerequisite for the automation of business activities across organizational boundaries. Electronic Business (e-Business) interoperability presents a difficult problem as partner organizations develop their systems independently, leading to differences in technology platforms, communication protocols, data formats, and information and business processes semantics. This problem becomes more difficult to resolve as the number of autonomous partner organizations grows, and business relationship become more complex. Consider, for example the travel industry business network illustrated in Figure 1. Because of the global nature of the travel industry, travel agencies have to manage business relationships with a large number of suppliers in diverse sectors across a world-wide business network. To increase their revenue and competitive opportunities, suppliers form alliances that allow them to cross-sell services of suppliers in other sectors. Through such alliance networks, airlines can provide additional services, for example allowing travelers

to book a hotel room and rent a car at the same time they book a flight with the airline. However, such complex business relationships have to be supported by interoperable e-business systems that facilitate information flows among a large number of disparate partner organizations.

Interoperability requirements of e-business applications have been studied extensively and classified into different types. For example, Bussler [1] classified interoperability into three levels: technical, information, and business process levels. Technical level interoperability deals with disparate communication protocols, language environments, and technology platforms used by business partners. Information level interoperability concerns business data heterogeneity, and can be further classified into: syntax, structure, and semantic heterogeneity [2]. Syntax heterogeneity refers to differences in formats used to represent data (e.g. tagged document formats, positional document formats, delimited document formats, etc.). Individual organizations often use different schemas with different structure of business documents to represent the same business data. Such schematic differences are referred to as structure heterogeneity. Semantic heterogeneity concerns the differences in the meaning of individual data items, and in practice is the most difficult to deal with. Business process level interoperability is concerned with collaborative activities between partners. Bussler classifies business processes into two types: public processes and private processes. A private process represents a flow of business tasks within an enterprise, while a public process represents a flow of interactions between business partners. Business process level interoperability focuses on public processes that define external actions that each participant performs, given a particular role that the partner assumes in the collaboration. All participants must have a common understanding of their joint public business processes including the meaning of each exchanged message, the flow of activities, and allowable actions and responses [3].

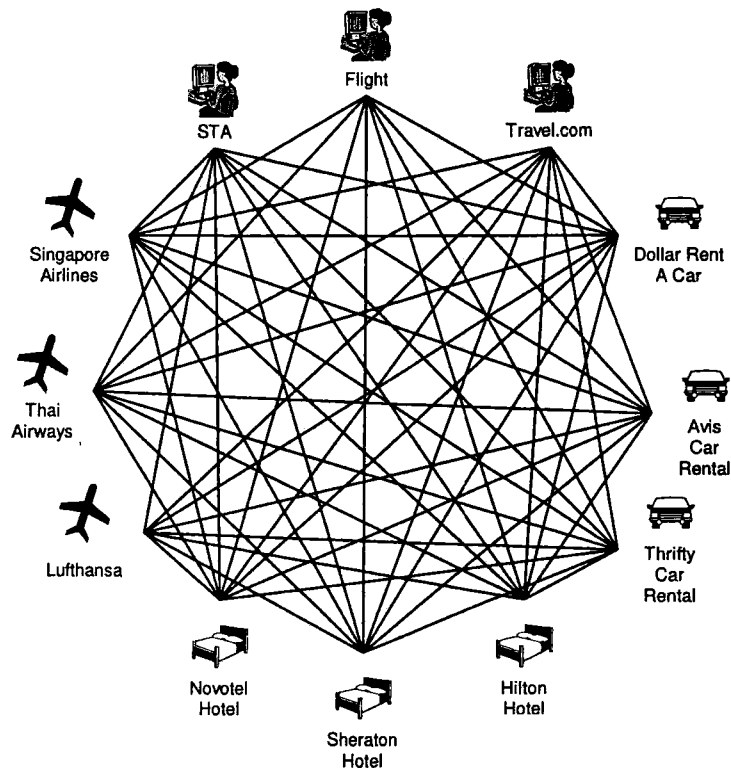


Figure 1. Travel Industry Business Network

In order to support e-business activities, partner companies must address the technical, information and business process interoperability issues described above. Numerous approaches, standards and technologies have been developed in an attempt to enable organizations of any size to conduct business electronically. Early e-business systems such as EDI [4], relied on direct connections between partner organizations using expensive proprietary technologies and private value-added networks (VANs), making such solutions unaffordable by smaller companies [5]. These technical interoperability issues have been recently addressed by deploying Internet infrastructure to provide inter-enterprise connectivity. The Internet provides ubiquitous public network with common protocols and standardized network infrastructure and significantly reduces the cost barrier for SMEs (Small and Medium sized Enterprises). Various Internet security protocols (such as Public-Key Infrastructure (PKI) [6], and Secure Sockets Layer (SSL) [7] have been developed to address security issues related to sending business information over a public network. Another significant recent

development was the emergence of Web Service as a universal technical level interoperability platform for e-business applications. Web Services address interoperability problems associated with applications written in different languages, running on different platforms [8]. Web Services use SOAP [9], an XML-based messaging protocol, widely supported by e-business systems and standards including BizTalk [10] and ebXML [11] and enables the delivery of documents as message payloads between disparate systems. An important feature of Web Services is that they can execute remote procedure calls (RPCs) providing a programmatic alternative to the document-centric model.

However, information-level and business process interoperability issues remain a significant problem in most e-business applications, particularly in environments characterized by a large number of autonomous partner organizations, as the case, for example, in the travel industry. While syntax heterogeneity has been addressed by acceptance of XML as a standard format for representing business data [12], semantic interoperability remains a large

unsolved problem. Understanding the information and business process semantics of partner systems is still a major challenge for organizations conducting business electronically [13].

In this paper we focus on information-level interoperability issues and discuss the opportunities presented by the service-centric approach to improve the interoperability of inter-enterprise applications. We first discuss the limitations of current document-centric approaches (section 2), and then contrast the document-centric and service-centric interoperability models (section 3). In the concluding section (section 4) we summarize the benefits of the service-centric approach to e-business.

2. Limitations of Existing e-Business Interoperability Approaches

Over the last 20 years a number of e-business approaches including EDI, BizTalk, RosettaNet [14], and ebXML have been developed and deployed in various industry domains. These approaches rely on the interchange of business documents as the interoperability mechanism and range in scope from point-to-point solutions to cross-industry standards as shown in Table 1.

Approach	Scope	Method
EDI	Point-to-Point	Translation-Based
BizTalk	Point-to-Point	Translation-Based (XML)
RosettaNet	Industry-Wide	Based on Standard PIPs
ebXML	Cross-Industry	Registry-Based

Table 1. Comparison of e-Business approaches

EDI is historically the first attempt to establish e-business interoperability standard and requires that the parties agree on the format and content of documents used for business communications [15]. Using EDI, each business partner pair has to standardize the messages that are used to conduct e-business. This inherently point-to-point solution does not scale well; each new business relationship requires a new set of message standards and

corresponding translation software that converts internal proprietary data formats into trading partner-specific standard messages. This leads to proliferation of diverse EDI versions that are established over time by dominant partners or groups of organizations. Existing EDI standards are based on X12, or EDIFACT which defines a limited set of common business messages (e.g. purchase order, invoice, etc.), that companies further refine to produce their own message formats. The creation of a new business messages or changes in the data types of the existing standard message sets are complex and time consuming activities, as the modifications have to be approved by the standards bodies that maintain the X12 and EDIFACT standards. This inflexibility limits the use of EDI to support new business domains, and new products and services.

More recent e-business standards use XML as the message syntax and XSLT to translate messages between partner organizations, avoiding the need for expensive translation software. For example, BizTalk Server is an XML-based system that provides a generic XML translation tool based on XSLT. However, similar to EDI, BizTalk is a point-to-point solution that requires human experts to understand the structure and semantics of the source and target XML documents, in order to define a translation map. Individual partner organizations need to create and maintain a mapping library for every partner they interact with, and this severely limits the scalability of the solution. Another e-business approach is exemplified by RosettaNet, an industry-wide standard that attempts to avoid point-to-point translations by explicitly defining standard semantics of business messages and processes. RosettaNet has been successful in the high-technology industry supply chain applications, but has not gained wide acceptance outside this industry sector. This success can be largely attributed to highly standardized business processes and data used in the high-technology industry domain, combined with commitment to implementation of standard-based solutions. ebXML is a cross-industry standards-based approach that supports dynamic partnerships via a registry that can be used to discover suitable partner organizations. Unlike RosettaNet, ebXML standard specifies e-business communication infrastructure, but it does not attempt to directly address information and business process heterogeneity. Such interoperability issues are left to each industry domain to address by developing industry-wide standards, or by adopting existing

standards in conjunction with ebXML infrastructure.

3. e-Business Interoperability Models

As noted in section 2 above, most existing e-business approaches use the document-centric interoperability model characterized by shipping business documents between partner systems. The main attraction of the document-centric approach is its ability to use platform-neutral business documents that do not require partner organizations to use compatible technology platforms. Business documents provide a level of abstraction that allows automation of inter-enterprise business processes based on mutual agreement about the structure and semantics of documents [16], [17]. However, document-centric e-business suffers from limited scalability and flexibility associated with extensive reliance on document translations or complex industry-wide document specifications. Scalability in this context refers to the ability of the solution to accommodate a large number of partner organizations with diverse business semantics without unduly increasing the complexity of the specification, and at the same time avoiding the need for point-to-point transformations. Flexibility concerns the need to allow the evolution of the standard specifications to accommodate changes in business processes and data semantics without impacting on existing applications [18]. The principal limitation of document-centric e-business is its use of large and complex message structures that implicitly represent complex business processes and contain embedded business rules [19]. The complexity of message structures arises from designing message payloads to include all the information needed to perform corresponding business functions without reference to information received in previous messages, making the interaction essentially *stateless*. For example, the OTA (OpenTravel Alliance, <http://www.opentravel.org>) Flight Booking Request document (OTA_AirBookRQ) contains flight booking details as well as payment details assuming that both sets of data are submitted at the same time. In practice, however, the two operations (i.e. flight booking and payment) are often performed separately resulting in duplication of information and potential data inconsistencies, as the OTA_AirBookRQ is used for both operations even when performed separately. Decomposing the flight booking request into two separate operations (i.e. flight booking and airline ticket payment) leads to simplification of the business process, improved

flexibility, and potential for reuse of the payment operation in another context (e.g. car rental, or a hotel room booking). While the stateless style of interaction reduces the number of messages needed to implement a particular business function, it results in complex and redundant data structures making changes to standard messages difficult to perform without producing undesirable side-effects that invalidate existing applications. In effect, the document payloads form the interface between applications and therefore introduce high levels of coupling and interdependencies by externalizing the document data structures. This makes document-centric standards difficult to develop and maintain, and expensive to implement. It can be argued that the requirements of today's dynamic e-business environment cannot be satisfactorily addressed using an interoperability mechanism based on document interchange.

3.1 Service-centric Interoperability Model

The emergence of Web Services as the standard platform for e-business applications addresses the technical level interoperability requirements. Web Services remove the need to use document interchange as a mechanism for application interoperability by providing a "homogeneous" application deployment environment irrespective of the underlying technology platforms (e.g. .Net, J2EE, etc.) used by individual partner enterprise applications. The service-centric model enables the incorporation of partner services into a client application via procedure calls making the implementation details of the service transparent. In effect, the application can be viewed as a *virtual* application operating transparently across multiple partner computing environments and incorporating (consuming) services published by different service providers.

Web Services enable service-centric e-business, with interoperability model based on well-defined service interfaces, reducing the problem of standardizing document formats and data semantics to a more manageable task of standardizing service interfaces for a given application domain. Such domain-specific service interfaces are conceptually similar to Application Programming Interfaces (APIs) that are used extensively in programming environments. Interoperability of service-oriented applications relies on stable service interfaces used consistently across the application domain (e.g. travel, healthcare, or education). Standardization of service interfaces across an industry domain ensures that service providers (e.g. airlines)

publish identical interfaces, avoiding the need to interpret the semantics of interfaces published by individual service providers. The abstraction level of standardized, domain-specific service interfaces is closely related to business requirements for a particular application domain; in effect, standard service interfaces constitute a high-level API for building domain-specific applications.

To ensure interoperability of e-business applications standard service interfaces must be well-designed using a methodology based on sound engineering principles [20]. The key difference between the document-centric and the service-centric interoperability models is that service interfaces can be designed to minimize interdependencies (i.e. coupling) between services by encapsulating message (data) structures and exposing method (operation) signatures that constitute a *stable* contract between the service provider and the service consumer. Data engineering principles can be applied to the design of service interfaces maximizing service cohesion and resulting in stable and maintainable e-business applications [21], [22].

4. Conclusions

With growing acceptance of service-oriented computing, and in particular Web Services as the preferred platform for the implementation of inter-enterprise applications, there is an urgent need to ensure that Web Services applications do not suffer from the same interoperability issues that characterize earlier, document-centric e-business approaches. There is some evidence that developers of Web Services applications favor the document-centric model in order to optimize performance over slow and unreliable networks. In many instances, Web Services infrastructure is used to ship structured documents as SOAP message payloads with little, or no attempt to take advantage of the service-centric interoperability model. For example, OTA message standard specification is used throughout the travel industry as the basis for Web Services implementations (e.g. Sabre (<http://www.sabre-holdings.com/>) is using Web Services to exchange OTA messages through its Global Distribution System gateway).

Unlike the message-based (document-centric) approaches Web Services can be used to implement business interactions as services over the Internet. Important advantage of the service-centric approach is that service interfaces can be designed to significantly limit the exposure of metadata, considerably improving the stability and robustness of e-business applications. Other advantages of the service-centric approach include

improved software reliability simplified development, and support for evolution of interfaces [23].

We note here that adopting the service-centric approach does not eliminate the need to agree on information and business process semantics, but it makes the problem more manageable by limiting the scope of agreement to service interfaces that correspond to simple business functions (e.g. payment). Well-designed service interfaces address the equally important problem of limiting exposure of complex and often redundant data structures that characterizes the document-centric approach and leads to high-levels of interdependency between applications. Moving away from an interoperability model based on document interchange and adopting a programmatic approach to e-business applications results in a higher level of abstraction associated with the use of application programming interfaces. As experience with programming APIs demonstrates, the benefits of standardized interfaces include improved software reliability, reusability, extensibility, and maintainability, and can ultimately lead to significant application development productivity gains.

The key requirement for achieving semantic interoperability in service-oriented applications is the standardization of service interface for a given application domain (e.g. travel). Without industry-wide interface standards, equivalent services published by different providers will not be compatible, placing a burden for resolving the inconsistencies on service consumers.

5. References

- [1] Bussler, C. 2001, 'B2B Protocol Standards and Their Role in Semantic B2B Integration Engines', *Bulletin of the Technical Committee on Data Engineering*, vol. 24, no. 1, pp. 3-11.
- [2] Ouksel, A. & Sheth, A. 1999, 'Semantic Interoperability in Global Information Systems: A Brief Introduction to the Research Area and the Special Section', *SIGMOD Record Special Interest Group on Management of Data*, vol. 28, no. 1. Available: <http://www.acm.org/sigmod/record/issues/9903/index.html/special/intro.pdf.gz>
- [3] Medjahed, B. et al. 2003, 'Business-to-Business Interactions: Issues and Enabling Technologies', *The VLDB Journal*, vol. 12, no. 1, pp. 59 - 85.
- [4] United Nations Economic Commission for Europe 2005, *UN/EDIFACT (United Nations Directories for Electronic Data Interchange for Administration, Commerce and Transport) Web Site* [Online]. Available:

<http://www.unece.org/trade/untdid/> [Accessed 3 June 2005].

[5] Adam, N. R. A., I.; Atluri, V.; Yesha, Y. 1998, 'EDI through a Distributed Information Systems Approach', in Proceedings of the Thirty-First Hawaii International Conference on System Sciences, vol. 7, pp. 354-363.

[6] The Internet Engineering Task Force 2004, Public-Key Infrastructure (X.509) Working Group [Online]. Available: <http://www.ietf.org/html.charters/pkix-charter.html>.

[7] Freier, A. O., Karlton, P. & Kocher, P. C. 1996, The SSL Protocol Version 3.0 [Online]. Available: <http://wp.netscape.com/eng/ssl3/ssl-toc.html>.

[8] Seely, S. 2002, Building Industry Standard WSDL [Online]. Available: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnservice/html/service02202002.asp> [Accessed 30 September 2003].

[9] Box, D. et al. 2000, Simple Object Access Protocol (SOAP) 1.1: W3C Note 08 May 2000 [Online]. Available: <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>.

[10] Microsoft Corporation 2005, Microsoft BizTalk Web Site [Online]. Available: <http://www.microsoft.com/biztalk/> [Accessed 3 June 2005].

[11] OASIS 2005, ebXML Web Site [Online]. Available: <http://www.ebxml.org> [Accessed 3 June 2005].

[12] Bray, T. et al. 2004, Extensible Markup Language (XML) 1.0 (Third Edition): W3C Recommendation 04 February 2004 [Online]. Available: <http://www.w3.org/TR/2004/REC-xml-20040204/>.

[13] Medjahed, B. et al. 2003, 'Business-to-Business Interactions: Issues and Enabling Technologies', The VLDB Journal, vol. 12, no. 1, pp. 59 - 85.

[14] RosettaNet 2005, RosettaNet Web Site [Online]. Available: <http://www.rosettanet.org/> [Accessed 3 June 2005].

[15] Kotok, A. 1999, XML and EDI Lessons Learned and Baggage to Leave Behind [Online]. Available: <http://www.xml.com/pub/a/1999/08/edi/index.html> [Accessed 3 April 2002].

[16] Glushko, RJ, Tenenbaum, JM & Melt, B 1999, 'an xml framework for Agent-based E-commerce', Communications of the ACM, vol. 42, no. 3, pp. 106-14, viewed March 1999, <http://doi.acm.org/10.1145/295685.295720>.

[17] Glushko, RJ & McGrath, T 2002, 'Document Engineering for e-Business', DocEng'02, ACM, McLean, Virginia, USA, pp. 42-8.

[18] Feuerlicht, G., Implementing Service Interfaces for e-Business Applications, The Second Workshop on e-Business, in conjunction with ICIS 2003, December 13-14, 2003, Seattle, USA, ISSN: 1617-9846

[19] Damodaran, S. 2004, 'B2B Integration over the Internet with XML: RosettaNet Successes and Challenges', in Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters (WWW2004), ACM Press New York, NY, USA, New York, NY, USA, pp. 188 - 195.

[20] Papazoglou, M.P. and Yang, J. (2002), Design methodology for Web services and business processes. In Proceedings of the 3rd VLDB-TES workshop (Hong Kong, August, 2002). Springer, pages 54-64.

[21] Feuerlicht, G, Designing Service-Oriented e-Business Applications using Data Engineering Techniques, The Third Workshop on e-Business, in conjunction with ICIS 2004, December 11, 2004, Washington D.C., USA, ISBN:957-01-9161-9

[22] Feuerlicht, G., Meesathit, S., Design Method for Interoperable Web Services, Proceedings of the 2nd International Conference on Service Oriented Computing, New York City, NY, USA, November 15-18, 2004, ISBN 1-58113-871-7, pages 299-307

[23] Bieber, G. & Carpenter, J. (2001), Introduction to Service-Oriented Programming (Rev 2.1) Available on: <http://www.openwings.org/download/specs/ServiceOrientedIntroduction.pdf>